TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MIRA VALKONEN

BIOIMAGE ANALYSIS USING CONVOLUTIONAL NEURAL NETWORKS AND FEATURE ENGINEERING

Master of Science Thesis

Examiner: Prof. Matti Nykter
Supervisor: Dr.Tech. Pekka Ruusuvuori
Examiner and topic approved by the
Faculty Council of the Faculty of Computing
and Electrical Engineering
on 1st February 2017

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Electrical Engineering
VALKONEN, MIRA: Bioimage analysis using convolutional neural networks and feature engineering
Master of Science Thesis, 45 pages
May 2017
Major: Biomeasurements and bioimaging
Examiner: Prof. Matti Nykter
Keywords: Image analysis, convolutional neural network, machine learning, deep learning, feature engineering, digital pathology

In digital pathology, analysis of histopathological images is mainly time-consuming manual labor and prone to subjectivity. Automated image analysis provides methods to analyse these images in a quantitative, objective, and efficient way.

The aim of this thesis study was to implement an image analysis pipeline to analyse histological images and quantitatively characterise tissue histology. To achieve this, a feature extraction library was implemented that utilises feature engineering and convolutional neural network to extract quantitative characteristics from histological images. Machine learning with supervised learning was then applied to train multiple discriminative models based on the quantitative histological characteristics.

The performance of the implemented system was evaluated in detection of breast cancer metastases in whole slide images (WSI) of lymph node sections. Three logistic regression models and a convolutional neural network were trained to separate metastatic tissue from normal tissue. Each model was able to detect the metastases with high accuracy (AUC = 0.89–0.97).

The results show that the implemented image analysis system provides an accurate detection of hot-spot regions in WSIs. These image analysis tools can be directly integrated into software that can reduce the workload of pathologists by guiding the pathologist to concentrate on regions of interest, and provides tools for quantification of tissue histology. These advancements will help to move towards the transition to digital pathology in clinical practice.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Sähkötekniikan koulutusohjelma
VALKONEN, MIRA: Biokuvien analysointi konvoluutioverkoilla ja piirteenirroituksella
Diplomityö, 45 sivua
Toukokuu 2017
Pääaine: Biomittaukset ja –kuvantaminen
Tarkastaja: Prof. Matti Nykter
Avainsanat: Kuva-analyysi, konvoluutioneuroverkko, koneoppiminen, syväoppiminen, piirteenirroitus, digipatologia

Digipatologiassa histologisten kuvien analysointi on usein työlästä, aikaa vievää ja patologin omakohtaiseen tulkintaan taipuvaista. Automaattinen kuva-analyysi tarjoaa tehokkaita menetelmiä histopatologisten kuvien kvantitatiiviseen ja objektiiviseen tulkintaan.

Tämän diplomityön tavoitteena oli toteuttaa menetelmiä histologisten kuvien analysoimiseen ja kudoksen kvantitatiiviseen esittämiseen. Tämän tavoitteen saavuttamiseksi toteutettiin menetelmä, jolla kuvadata voidaan muuntaa informatiivisiksi numeerisiksi piirteiksi. Menetelmä perustuu syviin konvoluutioneuroverkkoihin ja perinteiseen piirteenirroitukseen. Koneoppimisen menetelmin opetettiin useita luokittelijoita, jotka pystyvät erottelemaan erilaisia kudoksia piirredataan perustuen.

Menetelmän toimivuutta testattiin syöpäkudoksen ja terveen kudoksen erottamiseen histologisista kuvista. Luokitelussa käytettiin neuroverkkoa ja kolmea eri piirteillä opetettua regularisoitua logistisen regression mallia. Tulokset osoittivat, että menetelmä pystyi tunnistamaan erinomaisella tarkkuudella syöpäkudoksen normaalista (AUC = 0.89–0.97).

Tämän diplomityön osana kehitetyt kuva-analyysin työkalut on mahdollista yhdistää suoraan ohjelmistoon, joka voisi toimia digipatologiaa edistävänä työkaluna. Tällaisen ohjelmiston avulla pystyttäisiin pienentämään patologien työtaakkaa ohjaamalla patologia tarkastelemaan kiinnostavia alueita kudoskuvista ja tekemään diagnosoinnista objektiivisempaa.

PREFACE

This thesis work was conducted at BioMediTech, Institute and Faculty of Medicine and Life Sciences, University of Tampere for the completion of a Master's degree programme in Electrical Engineering at Tampere University of Technology, Finland. Firstly, I would like to thank professor Matti Nykter for providing the opportunity to work as a part of Computational Biology group and carry out this thesis study.

My sincere gratitude goes to my supervisor Pekka Ruusuvuori for all his advices and support during the various stages of this project. Also, I would like to give a special thanks to my colleagues in Bioimage Informatics team, Kimmo and Kaisa for their valuable insights regarding image processing, and Leena for providing deeper biological understanding. Additionally, I would like to give thanks to my other colleagues in the Computational Biology group for their general support.

Also, special thanks goes to Petteri for pushing me during the writing process and for having the time to listen to my whining about never finishing up. Finally, I deeply appreciate the support given by my family and friends during my years at TUT. I thought I could never say this, but here it is... It is DONE!

16 May 2017
Mira Valkonen

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

CNN	Convolutional neural network
HE	Hematoxylin and eosin
OD	Optical density
GLCM	Gray level co-occurrence matrix
LBP	Local binary pattern
HOG	Histogram of oriented gradients
glm	Generalized linear model
glmnet	Generalized linear model with elastic net regularization
ROC	Receiver operating characteristic
AUC	Area under the ROC curve
WSI	Whole slide image

1. INTRODUCTION

The last decades have gone through great advances in microscopic imaging of biological tissue. Advances in computational power has lead to the availability of significant amount of multi-dimensional bioimage data in almost every branch of biology [1]. This has revolutionized how biologists visualise molecular and cellular structures and study their functions. Now the major challenge is how to interpret such amount of bioimage data in a quantitative, objective, automatic, and efficient way. Automated image analysis provides solutions for this challenge [2; 3].

Image analysis in general aims at understanding an image by extracting meaningful information from the image. To extract this information, the image needs to be represented in an entirely objective and quantitative way. There are several image processing methods that can be applied to acquire a quantitative representation of an image. These methods have been applied in many different applications for decades. For example, satellite image segmentation [4], face analysis and identification [5], and industrial quality control applications [6], can be all addressed with similar image analysis methods.

In biomedical field, image analysis provides methods for turning bioimage data into useful biological information. These tools are increasingly applied also in digital pathology [7; 8]. Digital pathology is quite new, rapidly expanding field and is constantly generating more high-throughput bioimage data [9]. In digital pathology, whole-slide scanners are used to digitize glass slides containing tissue samples into high resolution images that can be viewed, managed, and analysed. Ideally, automated image analysis could already be used to screen the samples and provide regions of interest for pathologist review, or even proceed directly to diagnostic decisions. A more realistic scenario is to use automated image analysis for pre-screening the images in order to give supporting information and to exclude areas not relevant for diagnosis. Yet, in many clinics, analysing histopathological slides is still mainly time-consuming manual labor and also prone to misinterpretation and

subjectivity.

In digital pathology, automated image analysis system should be able to mimic the decision making process of a pathologist. There are certain decision rules that apply in the diagnosis process, and these type of rules can be replaced by a quantitative analysis of a numerical data. Supervised machine learning provides tools for achieving this goal [10]. Image analysis systems that rely on supervised learning derive decision rules based on labeled training data.

There are currently two main approaches to implement a supervised learning based image analysis system: traditional approach and neural network based methods. Traditional approach utilises feature engineering combined with machine learning, whereas, neural network models are mainly based on convolutional neural networks (CNN) and deep learning. A number of studies available in the literature show a great potential of traditional image analysis methods in analysing bioimages, such as, detecting metastasis from whole slide images [11], detecting early neoplastic changes [12], and survival prediction [13]. However, the deep learning based methods have recently gained ground in image analysis tasks also in biomedical imaging field. Deep learning term is used for neural networks with deep hierarchical structures and huge amount of trainable model weights. When a large amount of training data is available, deep learning can achieve high accuracy in many image analysis tasks [14; 15; 16]. Still, methodology for interpreting a deep classifier model is currently lacking, which is a disadvantage when one of the main goals is to convert bioimage information into biological information.

Deep CNN can be also used as a feature extractor, and this enables combining these two common approaches. This combination of deep CNN features and manually engineered features provides a powerful image analysis tool, as it would have the advantages of both methods: the high accuracy and the interpretability. A feature extraction library, including customized manually engineered features and deep CNN features, would provide useful tools to interpret bioimage data in a quantitative, objective, and efficient way.

The aim of this thesis study was to provide computational methods to automatically extract quantitative histological features from histological images. This was achieved by implementing an image analysis and feature extraction library. This library includes, customized feature extraction tools for digital pathology, as well as general image analysis tools for many other applications in histopathology. These image

analysis tools can be directly integrated into software that can reduce the workload of pathologists by guiding a pathologist to concentrate on regions of interest, and providing tools to quantify the full complexity of tissue histology and morphology. These advancements will help to move towards the transition to digital pathology in clinical practice.

This research was performed at BioMediTech, University of Tampere, and it is a part of three-dimensional (3D) histogenomic modeling of whole prostate project funded by TEKES. This developed feature extraction library will be further extended to three dimensions and integrated into 3D histogenomic model of prostate that is being currently developed. This project involves collaboration with The Molecular Biology of Prostate Cancer group at BioMediTech and a local company Jilab Oy.

The thesis is structured as follows. Second chapter will discuss the theory of image analysis and consider more closely the methods utilised in this research. The third chapter will present the implemented image analysis pipeline, and the materials used in this study. The fourth chapter will present the study results which are then discussed in the chapter five. And finally, the sixth chapter will conclude the work.

2. IMAGE ANALYSIS

Human can intuitively without any effort recognise objects in different surroundings, from various distances and angles. Human can also understand the surroundings and the context, and based on that, make a reasonable guess about what has happened or what is going to happen. For a machine to reach even nearly as high-level of comprehension and detection accuracy, it would require complex algorithms and significant computing power. Human visual system is, therefore, a great example of an image analysis system, and this is the reason why many image analysis methods are inspired by human visual cortex. [17, p.1-2]

Image analysis can be defined as transformation of an image into a meaningful information. This information is based on quantitative data that has been extracted from an image using digital image processing methods. The difference between image analysis and plain image processing is that digital image processing has an image as an input and output, whereas image analysis has numerical or conceptual output. [18, p.419]

There are two common approaches to implement a learning based image analysis system: traditional feature based approach and neural network based approach. These approaches differ from each other to some extent, nevertheless, both of them consists of three fundamental steps of image analysis system: preprocessing, feature extraction, and image interpretation [17, p.5-9]. Preprocessing step includes all the processing that is performed to get the image data ready for the analysis, for example image enhancement. The main goal of a feature extraction step is to get a quantitative representation of the image. And finally, for a learning based image analysis system, the image interpretation step is commonly performed with a classification model trained with the quantitative feature representation.

In traditional image analysis, these three main steps are distinctly separable. This type of image analysis system rely on expertise and knowledge on the important discriminative properties for a certain application, and methods to manually

engineer tools to extract these discriminative features. These quantitative features can then be used to teach a machine learning algorithm to discriminate the relevant feature information [17, p.381-404]. Quantitative feature representation combined with machine learning methods provides powerful image analysis tools. Advantage of these type of models is that they are easy to interpret, and thus, the engineered features can be connected to the biological information. These traditional methods have been used for years in digital pathology and in many different applications, such as, cell detection and classification [19], grading prostate cancer [20], and identification of immune cell infiltration [21]. Additionally, these traditional approaches have been successfully used for quantitatively describing characteristics of histology [22; 12; 23]. However, previous studies have left room for improved feature engineering and classification performance.

Neural network based methods, on the other hand, do not need any specific feature engineering, because these methods have feature extraction and classification steps combined into one step [17, p.404-410]. These type of methods are based on deep neural network architectures, like deep CNN consisting of vast amount of trainable model weights [24, p.3-7]. Supervised learning can be used to learn both the feature representation as well as the classification model by providing only input image and desired output. In recent years, these methods have become the state-of-art methods in many bioimage classification and detection tasks [25; 26; 27]. When a large amount of training data is available, these type of models can achieve even the same level of classification accuracy as human professionals [28]. However, while the performance is top level, the model interpretation and link to biological information is somewhat undefined.

Besides classification, deep CNN can be also used for extracting features. This provides the opportunity to combine these two main approaches. The accuracy of deep learning combined with engineered features and machine learning would provide a powerful and interpretable image analysis tool. With known network architecture, these deep CNN feature representations could be better understood and linked with spatial information that they represent. There are studies that have established the benefits of combining these two methods, and achieving even better performance than either of the methods alone [29; 30; 31]. Also, when CNN is used as feature extractor, the size of dataset is not an issue, as pre-trained networks can be used with fine-tuning or transfer learning.

Three fundamental steps of an image analysis system are presented in the following sections in the light of this study. Preprocessing section includes some of the common preprocessing steps that can be applied to bioimages. Feature extraction section presents the extraction of feature representation using traditional feature engineering and a deep CNN. Finally, the classification section presents basics of machine learning, and more closely, one approach that can be applied to perform the final image interpretation step of image analysis.

2.1 Preprocessing

Preprocessing steps may vary a lot between different datasets and applications. Preprocessing includes all the image processing steps that are needed to get the data ready for the analysis. The aim of preprocessing is to improve and enhance the quality of image data and remove unwanted noise and distortions that could have an effect on the analysis. With preprocessing, comparable feature data can be generated for images with varying quality or even between independent image datasets. [17, p. 56]

Histopathological image analysis has its challenges. The image quality can be affected by multiple factors, since the tissue sample preparation is mainly manual work. This preparation can cause variation in tissue section thickness, which can again, affect the staining and final color of the image. In addition, when the sample is scanned into digital form, factors such as, the used scanner, scanner configurations, and image compression can affect the final quality. As this work is focused on histopathological image analysis, the following section will discuss some basic preprocessing steps that are commonly applied to histological images (Figure 2.1).

2.1.1 Histogram matching

Correction of color variation is a common preprocessing step when analysing histological images. The colorspace normalisation throughout the whole image stack enables extraction of comparable feature data. Histogram matching is one possible approach for applying color correction. Histogram matching algorithm generates an output image based upon a specified reference histogram. This is achieved by

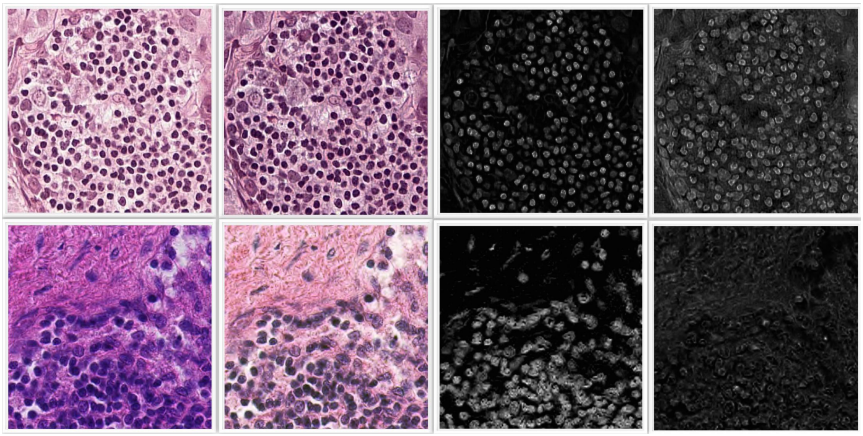

Figure 2.1 *Common preprocessing steps for histological images. Upper row presents the histogram matching step and color deconvolution for one sample image, and the lower row presents the same steps for another sample image.*

computing a mapping function that matches the input histogram to the reference histogram. [32, p.94-102]

To compute the mapping function, cumulative histograms of input image and the reference image are needed. If z and r present the gray levels in input image and in reference image, and their corresponding probability density functions are $p_z(z)$ and $p_r(r)$, the cumulative histograms can be presented as follows:

$$c_r = \sum_{k=1}^{L} p_r(r), \;\; k = 1, 2, .., L \tag{2.1}$$

$$c_z = \sum_{k=1}^{L} p_z(z), \;\; k = 1, 2, .., L \tag{2.2}$$

where L is the number of discrete gray levels in the digital image. The mapping function transforms each pixel intensity in the input image in a way that the result histogram matches the reference image, $c_r(k) = T(c_z(k))$. The mapping function can be defined by choosing gray level transformation T to minimize the absolute error between the gray level intensities of the reference histogram and the matched histogram:

$$\min |T(c_z(k)) - c_r(k)| \tag{2.3}$$

For a RGB image, the mapping function is computed for all three color channels.

2.1.2 Color deconvolution

Color deconvolution [33] can be used to convert red, green and blue channels of a RGB image into different biological stains. In biological applications of image analysis, this is usually an important preprocessing step, since different stains mark different structures within biological tissue. One of the most common example is hematoxylin-eosin (HE) stain, where the hematoxylin (blue) mainly stains the cell nuclei and eosin (magenta-red) stains the cytoplasm. This differential staining functions as an indicator of the amount of certain structures within the biological tissue based on the stain concentration. As these biological tissue samples are scanned into digital images, the attenuation of transmitted radiant power determines the optical density (OD) of the sample. According to Beer-Lambert law [34] this optical density, or more closely the absorbance of the sample, is exponentially proportional to the stain concentration. Each stain can be characterised by a specific absorption factor c. This is described by Beer-Lambert law as follows:

$$I_C = I_{0,C} * \exp(-A * c_C), \tag{2.4}$$

where A is the amount of stain with absorption factor c, $I_{0,C}$ is the intensity of light entering the tissue sample, and I_C is the intensity of light detected after passing the sample. Intensities present the gray value intensities in each RGB channel. As can be seen from the equation, the gray value intensities depend on the concentration of stain in a non-linear way [35]. Therefore, the RGB intensities cannot be directly used for the stain separation. However, optical density for each channel is linear with the concentration of absorbing material and that can be defined as follows:

$$OD_C = -log_{10}(\frac{I_C}{I_{0,C}}) = A * c_C \tag{2.5}$$

The stain-specific OD values that were used in this thesis study are presented by

Ruifrok et al. [33]. In their study, the stain-specific OD values were determined by measuring relative absorption for red, green and blue on slides stained with a single stain. An example of the OD matrix for the combination of hematoxylin and eosin:

$$\begin{matrix} R & G & B & \\ \begin{bmatrix} 0.18 & 0.20 & 0.08 \\ 0.01 & 0.13 & 0.01 \\ 0 & 0 & 0 \end{bmatrix} & & & \begin{matrix} Hematoxylin \\ Eosin \\ Background \end{matrix} \end{matrix} \tag{2.6}$$

Here, each row represents a specific stain, and each column represents the optical density as detected by the red, green and blue channel for a sample stained with single stain.

For separating the independent information of stain, an orthonormal transformation is performed. To get correct balance of the absorption factor for each stain, the transformation is normalized by dividing each OD vector by its total length. If C is a 3 by 1 vector for amounts of the three stains at a particular pixel, then the vector of OD levels detected at that pixel is $y = CM$. Where M is the normalized OD matrix. So to get the C, the OD image needs to be multiplied with inverse of the normalized OD matrix, which can be defined as the color-deconvolution matrix D. This results in orthogonal representation of the stains forming the image: $C = M^{-1}y = Dy$.

2.2 Feature extraction

The most important part of image analysis is to acquire the numerical representation of an image, since often the image analysis results are as good as its quantitative image representation. An ideal feature extractor would generate a feature representation that makes the classification problem trivial. Then again, much more complex classifier is needed if the feature representation is multidimensional and diverse. [36, p.12]

One method to present an image in a quantitative form, is to present it as a vector of an engineered set of features. Where each value in the vector presents some desired characteristic or property of the image. This feature value can be extracted from the whole image or just a small block of the image, which sums up to a set of vectors

per whole image. Expertise and prior knowledge is needed to capture the desired information in a numerical form, in a way that it differs from everything else in the image. Depending on the application, the selection of feature detector can vary. For example, if the desired information is related to a repeating texture or a shape, features related to frequency transform of the image might be suitable. Or if the desired information is related to brightness of the image, a mean intensity would be a good feature. However, the more incoherent and varying the wanted characteristic is, the more complex features are needed.

Another method is to use convolutional neural network to detect the important characteristics. Using neural network as a feature detector, no prior knowledge is needed about the important discriminative features. CNN combined with supervised learning can learn the discriminative feature representation of the image. The next subsections will explain in more detail these two approaches applied to the task of generating a quantitative feature representation of an image.

2.2.1 Feature engineering

There is no exact definition on what can be counted as an image feature but generally feature is a primitive characteristic or an attribute of an image. From image analysis point of view, a feature is a piece of information that is relevant for solving the task related to image classification. By a rough division, features can be divided into two groups. There are natural features that are apparent from the image and then there are artificial features that can be extracted from the image via some manipulation operation. Features such as mean gray value that represent the luminance is a natural features as it can be extracted directly from the image pixel values. Then again features related to frequency spectra is an artificial feature as it is extracted by first computing a frequency transform of the whole image. [18, p.535-537]

The feature concept is very general and therefore the set of extracted features can be highly dependent on the application at hand. In this study, the extracted engineered features include mainly texture features and descriptors related to specific keypoint structures such as points or edges in the image. Textures are complex visual patterns that have similar characteristics such as brightness, size, shape, density or other group property. These similarity groupings in an image give rise to texture features such as frequency, regularity, roughness, linearity or smoothness of the whole texture. The following subchapters present more closely the features that were used in this

study.

Statistical texture features

Statistical texture features include for example features derived from the image histogram and co-occurrence matrix. The derived features can be divided into first-order, second-order, and higher-order statistical texture features, where the order refers to the number of pixels that are used in defining this local feature. [37]

Image histogram presents the distribution of single pixel intensity values in an image. The first-order features can be derived from this gray level distribution to characterise the texture. The most often used feature parameters derived from histogram are mean, variance, standard deviation, skewness, and kurtosis. The variance and standard deviation are measures of the histogram width measured from the mean intensity value. With respect to image processing, skewness is a measure of asymmetry of intensity value distribution about mean intensity value of the image. Skewness can get negative or positive values. Positive skewness of an image histogram would mean that the gray level intensity image contains more darker shades, whereas, negative skewness stands for relatively more bright shades. Kurtosis is a measure of peakedness or tailedness of the pixel intensity distribution. Considering image data, high value of kurtosis would mean that most of pixels in an image are close to some same mean intensity value. [37; 38]

Second-order statistical features are based on the joint probability distributions of pairs of pixels and these features normally achieve high discrimination between different textures. A human being can spontaneously discriminate textures that differ in second-order moments and this seems to indicate that the statistics up to the second-order may be the most important ones in texture features [39]. The most popular second-order statistical texture features are derived from gray level co-occurrence matrix (GLCM [40]) that describes the spatial relations of similar gray tones. The matrix is computed by defining a certain distance and direction and analysing pairs of pixels that posses this given distribution of gray level values [40]. In other words, a GLCM is a distribution of co-occurring gray values at a given offset over an image. The offset is defined by distance and angle from a certain gray level. The following set of equations define a subset of standard properties derivable from a normalized GLCM:

$$Contrast = \sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2 \tag{2.7}$$

$$Dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j}|i-j| \tag{2.8}$$

$$Homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{2.9}$$

$$Angular\ second\ moment(ASM) = \sum_{i,j=0}^{levels-1} P_{i,j}^2 \tag{2.10}$$

$$Energy = \sqrt{ASM} \tag{2.11}$$

$$Correlation = \sum_{i,j=0}^{levels-1} P_{i,j} \left[\frac{(i-\mu_i)\ (j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \tag{2.12}$$

Where P is the GLCM and level is the max gray intensity value in the image. These GLCM properties can be used as numerical features of image texture that describe the gray level pixel arrangements.

Local Binary Patterns

Local binary patterns (LBP [41]) are theoretically straightforward and an efficient approach for extracting gray level and rotation invariant texture features from an image. Certain uniform local binary patterns are fundamental properties of local image texture and their occurrence histogram extracted with LBP operator has proven to be very powerful set of texture features [42]. LBP operator is a measure of the spatial structure of local image intensities. The basic idea of the LBP operator is to transform a local circular neighborhood into a binary pattern by thresholding the neighborhood with the gray value of the center pixel. Due to this thresholding, the features are robust in terms of gray level variations. The circularly symmetric neighborhood is determined by assigning parameters that control the quantization of the angular space and radius of the neighborhood. Example sets of circularly symmetric neighborhoods are presented in Figure 2.2 for three different parameter sets.

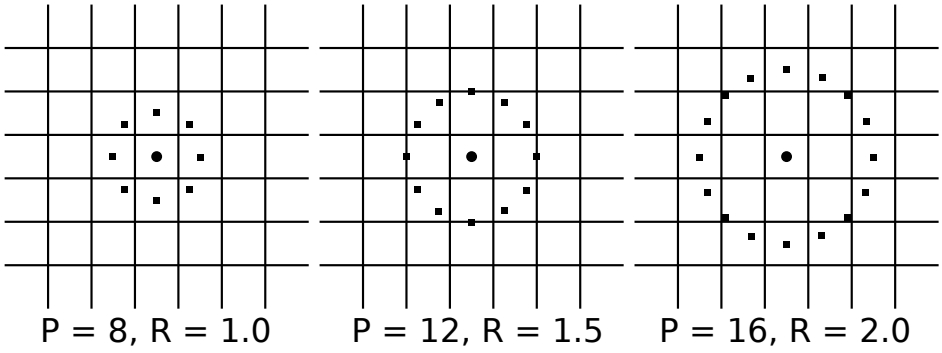

Figure 2.2 *Set of circularly symmetric neighborhoods determined by three different parameter pairs of radius and angular spacing.*

Certain binary patterns have proven to provide vast majority of all 3x3 patterns in textures. These are called uniform pattern as they have uniform circular structure in common, which contains very few spatial transitions. These uniform patterns are presented in Figure 2.3, where the circular binary pattern gets value of 1 if the value in circular neighborhood is greater than the value of center pixel, otherwise the pattern gets value 0. By computing the response image of LBP operator and computing the uniform pattern occurrence histogram from this response, structural and statistical approaches to feature extraction can effectively combined [42]. This way, the local binary pattern detects microstructures, such as edges, lines, spots, and flat areas, whose underlying distribution is estimated by the histogram.

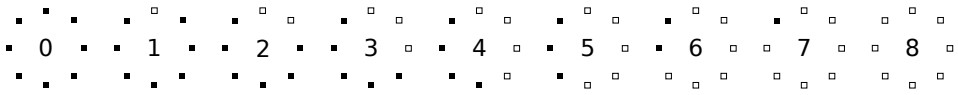

Figure 2.3 *Nine uniform patterns that provide vast majority of all 3x3 patterns found in textures. The circular binary pattern gets value of 1 if the value in circular neighborhood is greater than the value of center pixel, otherwise the pattern gets value 0.*

HOG: Histogram of oriented gradients

Histogram of Oriented Gradients (HOG [43]) is an edge and a gradient based descriptor. The main idea of this method is that the appearance and shape of an object can be characterised with the distribution of local gradients and edge

directions without knowing the positions of the gradients and edges. The processed image is divided into smaller windows and the distribution of local gradients and edge directions are calculated for each window.

Daisy: A Local Dense Descriptor

Daisy is an efficient dense descriptor for extracting local features from an image [44]. Daisy descriptor is a combination of Scale-invariant feature transform (SIFT [45]) and Gradient Location and Orientation Histogram (GLOH [46]) descriptor. The Daisy descriptor is mainly based on computing Gaussian convolutions.

SIFT and GLOH descriptors are methods for extracting keypoints that are invariant to translation, rotation and rescaling. These descriptors are based on histograms of gradient directions in local spatial regions. Each pixel in the local region contribute to the orientation and the norm of the gradient. The goal of a Daisy descriptor is to efficiently compute the gradients at every pixel location. This is achieved by replacing the weighted sums of gradient norms in the histogram generation by convolutions of the gradients in specific directions with several Gaussian kernels. The used Gaussian kernels are circular with increasing standard deviation values. For all the pixels, the neighborhood with certain radius is divided into a series of intersecting circles where the histograms of gradient directions are calculated. For each pixel position, the full descriptor is built as a concatenation of all the histograms of the intersecting circles.

2.2.2 Convolutional neural network as a feature extractor

Neural network is an interconnection of simple processing units called neurons. These fundamental parts of neural networks can be modeled with three basic elements: connecting links characterised by weights, summing operation of these weighted inputs, and activation function [47, p.32]. Figure 2.4 presents a model of an neuron, and the same neuron can be presented with the following equation:

$$y_k = f(\sum_{j=0}^{m} w_j x_j + b_k) \tag{2.13}$$

where $(x_1, x_2, .., x_m)$ are the input signals, $(w_1, w_2, .., w_m)$ are the weights of the neuron k, and y_k is the output signal of the neuron. b_k is the bias term that can be used to apply an affine transformation to the sum of weighted inputs. Activation function f defines the neuron output based on the sum of weighted inputs.

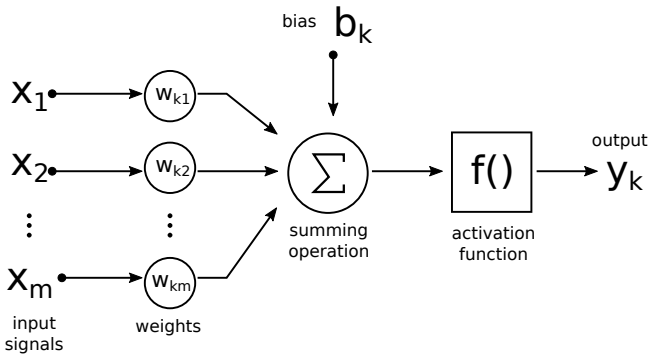

Figure 2.4 *Neuron, the fundamental part of neural network, modeled with three basic elements: connecting links characterised by weights, summing operation of these weighted inputs, and activation function.*

In a neural network, multiple neurons are organised as layers. Multilayer neural network has one or more hidden layers. Hidden layers are formed by neurons that interact between output neurons and input signal. By adding more hidden layers, neural network can extract higher-order statistics.

Like a neural network, human visual and cognitive system is composed of layers of interconnected neurons. The first model of visual cortex [48] presents cat's visual system as combination of simple and complex operations. This classical simple-to-complex model of visual cortex was inspiration to an hierarchical, multilayered artificial neural network called Neocognitron [49]. This Neocognitron has been the first step and inspiration in development of convolutional neural networks, which are widely used in today's image analysis applications.

Convolutional neural network

Convolution is a mathematical operation of two functions, that is commonly used in image processing. Convolution operates on two signals, one is the input signal, and the other one is filter. The output signal is produced by sliding this filter signal over the input signal. When processing one dimensional (1D) signals, convolution of f and g signals can be defined as follows:

$$(f * g)(i) = \sum_{j=-\infty}^{\infty} g(j)f(i-j) \tag{2.14}$$

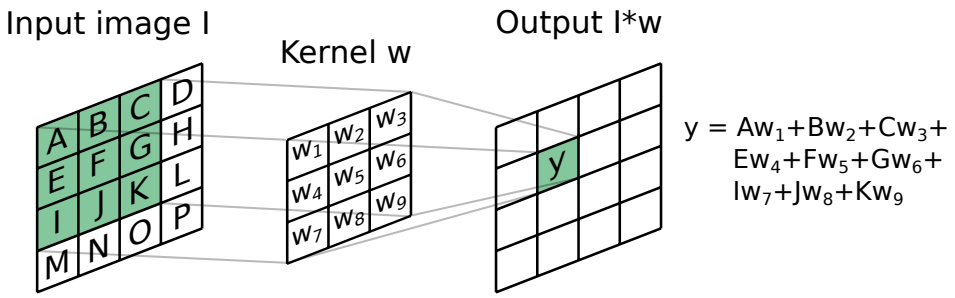

Figure 2.5 *Convolution operation applied to an image.*

Convolution can be applied also to two dimensional image signals. In image analysis, the input signal is two dimensional array and convolution is applied over more than one axis, so therefore, two dimensional kernel is used as filter. Similarly as in 1D case, for each position of the kernel, the overlapping values of the kernel and the input image are multiplied and summed. This operation is visualised in Figure 2.5.

Convolutional neural networks are neural networks that use convolution in the place of matrix multiplication at least in one of their layers. This means that that each neuron takes its synaptic inputs from a local receptive field in the previous layer. Receptive field presents the summed and weighted inputs, where the weights are the kernel weights. Kernel size defines the size of local receptive field. This output of one neuron becomes a extracted local feature. When this operation is applied for the whole input image, CNN can be used to generate map of local features for the whole image. [24, p.330-358]

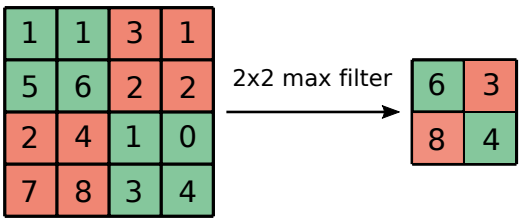

Figure 2.6 *Max pooling operation using 2 x 2 kernel.*

Convolution operation is usually followed by subsampling. The idea of this

subsampling is inspired by the classical simple-to-complex model of visual cortex. Complex operation, convolution, is followed by simple operation, subsampling. The function of this pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Max pooling is usually used as this subsampling operation. This operation is visualised in Figure 2.6. Max pooling operation is also applied locally and therefore the spatial features are preserved. Figure 3.3 presents a visualisation of convolutional neural network architecture with four convolutional layers followed by max pooling layers. [24, p.330-358]

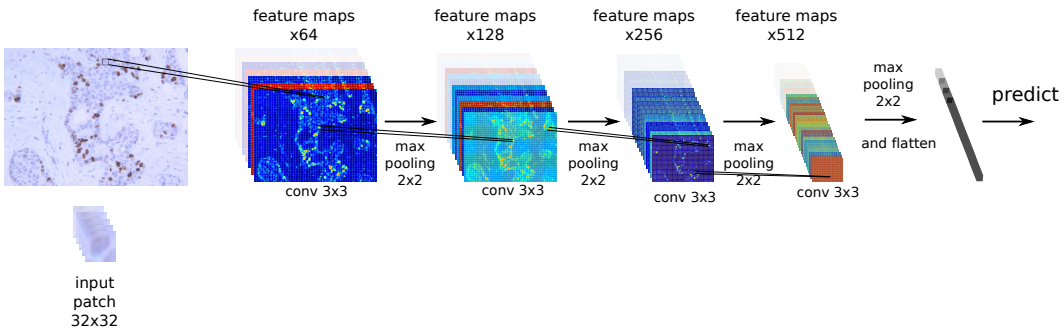

Figure 2.7 *Visualisation of convolutional neural network architecture with four convolutional layers followed by max pooling layers.*

Any of these feature maps within the network could be used as CNN features. When more layers are added in the network the final feature map will represents higher-order features. However, with each added layer the CNN architecture gets deeper, which results in increased amount of trainable kernel weights.

2.3 Classification and image interpretation

Classification is the final step in the image analysis process, and this step derives the final information and interpretation of the image. Classification is a process of identifying an object and defining a label or a class for it, based on its properties. The objects within a certain class possess similar properties. Machine learning can be used to build a classification model.

For a model to understand and classify an image, it has to be able to learn properties

of the image. This can be achieved by utilising machine learning methods. Most machine learning algorithms can be divided into two categories: supervised learning algorithms and unsupervised learning algorithms.

Broadly, unsupervised learning methods divide data into naturally separable groups and define discriminative model based on hidden structures in the data. Whereas, supervised learning algorithms use labeled training data to train a discriminative classification model. Supervised learning algorithm adjusts the model parameters based on error measure of the model output and the ground truth training data. The following subsections will look more closely into supervised learning methods in machine learning.

2.3.1 Building a machine learning model

By definition, machine is learning when the performance of running a certain task improves with experience [50]. The experience in this case is the presented training data. Ideally, with each new training sample that is presented to the machine, it would perform certain task better. Machine learning can be used to computationally learn a discriminative image classification model from the quantitative feature representation of an image.

The design of a machine learning algorithm can be divided into four main parts: specification of a dataset and the problem, a loss function, an optimization procedure, and a model. These components can be selected mostly independently from each other, which results in a wide variety of algorithms that can be obtained.

Loss function maps the learning process into a real value in a way, that it presents the performance of the algorithm. As the main goal of the learning process is to get better performance, the learning process loss have to be minimized. For example in classification problem, loss function presents the error of an incorrect classification. In other words, loss function maps an input instance to the difference between predicted label and true training label. Loss function parameters are the trainable parameters of the model, and therefore, it is usually used for estimating optimal parameters for a classification model. Optimal parameters of a model can be found by minimizing the loss function. [51, p. 40-48]

Loss function can be minimized through an optimization procedure. With each input instance of training data, the parameters of loss function are updated according to

optimization procedure. As the loss gets smaller during optimization, the model performance gets better. [51, p. 40-48]

The following section presents the machine learning algorithm that was used in this study to solve the image interpretation and classification problem.

2.3.2 Generalized linear model for regression

The goal of regression is to predict value of a target variable y given the value of a N-dimensional vector x of input variables. In linear regression, the target value is expected to be a linear combination of the input variables.

$$y(\mathbf{x}, \beta) = \beta_0 + \beta_1 x_1 + ... + \beta_N x_N \tag{2.15}$$

The vector of coefficients $\beta = (\beta_0, \beta_1, ..., \beta_N)$ define the regression model, that estimates the relationship between the target and the input variables. The main property of this type of model is its linearity, however, this factor sets significant limitations on the model. Linear model can only determine linear relationships between the target and the input variables. To determine nonlinear relationships, a nonlinear regression model must be used. [51, p. 137-147]

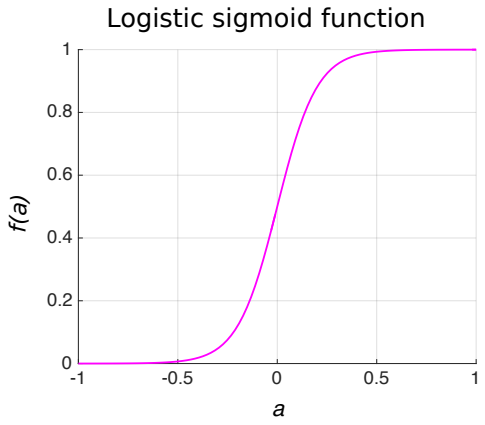

Figure 2.8 *Logistic sigmoid function.*

For a classification problem, instead of predicting a label for new sample, class-conditional probability can be estimated. To achieve this, linear model is generalized by applying nonlinear logistic sigmoid function (Figure 2.8). Logistic

regression is a commonly used model for binary classification problem, because it maps values from $[-\infty\ \infty]$ to the range [0 1]. This way, the model output can be directly used as posterior probability of a class. [51, p. 205-206].

The posterior probability of a class can be written as a logistic sigmoid acting on a linear function of the feature vector as follows:

$$P(class|x) = y(\mathbf{x}, \beta) = f(\beta^T \mathbf{x}), \tag{2.16}$$

Where $f()$ is the logistic sigmoid function:

$$f(a) = \frac{1}{1 + e^{-a}} \tag{2.17}$$

Logistic regression model represents the class-conditional probabilities as a linear function of the predictors:

$$P(class = 0|x) = \frac{1}{1 + e^{(\beta_0 + x_i^T \beta)}} \tag{2.18}$$

$$\begin{aligned} P(class = 1|x) &= \frac{1}{1 + e^{(\beta_0 + x_i^T \beta)}} \\ &= 1 - P(class = 0|x) \end{aligned} \tag{2.19}$$

Which can be presented as log-odds ratio:

$$\log \frac{P(class = 0|x)}{P(class = 1|x)} = \beta_0 + x_i^T \beta \tag{2.20}$$

The model coefficients $\beta = (\beta_0, \beta_1, ..., \beta_N)$ for linear regression model are estimated using maximum likelihood approach. For a dataset $\beta^T, f(\mathbf{x})$, the likelihood can be written as:

$$\prod_{i=1}^{N} P(class_i|x_i) \tag{2.21}$$

Or alternatively for computational convenience, as logarithm is a monotonic function maximum likelihood can be written as log-likelihood:

$$log \prod_{i=1}^{N} P(class_i|x_i) = \sum_{i=1}^{N} logP(class_i\|x_i) \tag{2.22}$$

In this study, logistic regression model with elastic net regularization was used (glmnet). This model is fitted by minimizing the negative binomial log-likelihood. This loss function for the model can be written as:

$$\min_{\beta_0,\beta} - \left[\frac{1}{N} \sum_{i=1}^{N} y_i * (\beta_0 + x_i^T \beta) - \log(1 + e^{(\beta_0 + x_i^T \beta)}) \right] + \lambda P_\alpha(\beta) \tag{2.23}$$

Where y_i presents the log-likelihood, N is the number of observations, $P_\alpha(\beta)$ the elastic net regularization term, and λ controls the overall strength of the regularisation. [52]

The regularization term is added to manage overfitting of the model. Elastic net regularization [53] is a combination of ridge penalty and lasso. The ridge penalty is based on shrinking the coefficients of correlated features towards each other while the lasso regularization is based on picking only one of the correlated features and discarding the others. The ratio of lasso and ridge in elastic net regularization is controlled by parameter α. When $\alpha = 1$ the regularization uses only lasso and when $\alpha = 0$, only ridge is used. The elastic net regularization can be written as:

$$P_\alpha(\beta) = (1 - \alpha) \left\|\beta\right\|_2^2 /2 + \alpha \left\|\beta\right\|_1 \tag{2.24}$$

A quadratic approximation of the log-likelihood is used, which results in penalized weighted least-squares problem. Finally, coordinate descent is used as an optimization procedure for the model weight update by minimizing the weighted least-squares problem of loss function. To find a local minimum of a function,

the coordinate descent method minimizes the function along one direction at each iteration. [52]

2.4 System evaluation

In classification problems, sensitivity and specificity are used to measure the performance of a binary classifier. Sensitivity (true positive rate) measures the proportion of positive samples that are correctly classified as positive. Specificity (true negative rate) measures the proportion of negative samples that are correctly classified as negative. These can be calculated as follows:

$$Sensitivity = \frac{number\ of\ correctly\ classified\ positive\ samples}{number\ of\ true\ positive\ samples} \tag{2.25}$$

$$Specificity = \frac{number\ of\ correctly\ classified\ negative\ samples}{number\ of\ true\ negative\ samples} \tag{2.26}$$

Precision (positive predictive value) measures the correct positive results divided by the number of all positive results. This measures the fraction of retrieved samples that are relevant considering the classification problem and it is defined as follows:

$$Precision = \frac{number\ of\ correctly\ classified\ positive\ samples}{number\ of\ positive\ classified\ samples} \tag{2.27}$$

F-score is also commonly used measure of binary classification accuracy. F-score considers the precision and sensitivity (also known as recall) of the classifier. It can be interpreted as a weighted mean of these two measures.

$$F - score = 2 * \frac{precision * recall}{precision + recall} \tag{2.28}$$

Receiver operating characteristic (ROC [54]) curve is another common method for analysing the performance of a binary classifier. The ROC curve, is a graphical plot that visualises the performance of a binary classifier as its discrimination threshold is varied. The curve is created by plotting the sensitivity against the false positive rate (1-specificity) at various threshold settings. This visual measure is usually

combined with area under the curve (AUC) measure that gives a numerical value for the classifier performance.

3. RESEARCH MATERIALS AND METHODS

3.1 Objective

The aim of this thesis study was to implement an image analysis pipeline for analysing histological images and quantitatively characterise tissue histology. To achieve this, a feature extraction library was implemented that utilises feature engineering and deep CNN to automatically extract quantitative characteristics from histological images. Machine learning with supervised learning was then applied to train multiple discriminative models based on these quantitative histological characteristics.

3.2 Bioimage dataset

The dataset used in this study consists of two independent hematoxylin and eosin stained whole slide image (WSI) datasets, 270 WSIs in total. Both datasets were provided for Camelyon16 challenge [55]. This challenge was organized in conjunction with the 2016 IEEE International Symposium on Biomedical Imaging (ISBI-2016). Figure 3.1 presents samples from both of the datasets.

The first dataset consists of 170 WSIs of sentinel lymph node sections collected at the Radboud University Medical Center (Nijmegen, the Netherlands). A total of 100 WSIs presented normal lymph node sections and 70 WSIs contained breast cancer micro- and macro-metastases. Altogether 60 of these cancerous lymph node sections were fully annotated and 10 only partially annotated. The second dataset of 100 WSIs was collected at the University Medical Center Utrecht (Utrecht, the Netherlands) and it contains 60 WSIs of normal lymph node sections and 40 WSIs with lymph node metastases. Of the 40 cancerous slides, 37 were fully annotated and 3 partially annotated. The whole slide images and the corresponding annotation masks were provided as multi-resolution pyramids in Philips BigTIFF

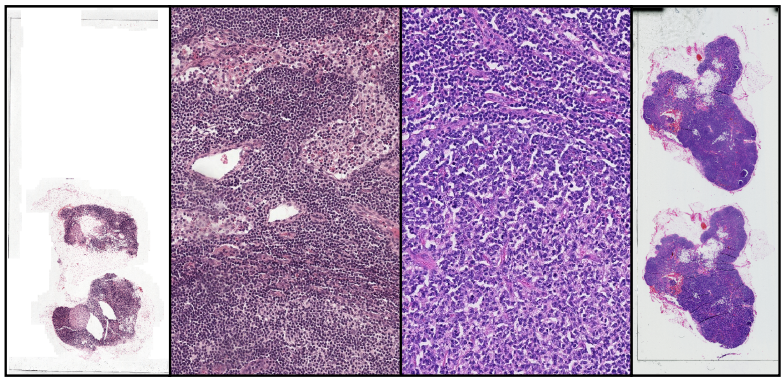

Figure 3.1 *Visualization of whole slide images from two different independent datasets used in this study. The images present hematoxylin and eosin stained lymph node sections.*

format. The pixel size of the images at the full resolution level was 243 nm. Only the fully annotated slides were used the to obtain both positive and negative training examples. The partially annotated slides were only used to obtain positive samples to avoid the risk of using unannotated metastatic regions as negative training data.

Generation of a fully annotated dataset with this extent requires a significant amount of manual work and expertise of pathologists. Thus, for image analysis method development and validation purposes this type of dataset is extremely valuable. To denote the size of the used dataset, uncompressed WSI file size can be 2-10 GB and the size of one WSI in pixels can be approximately 30 000-100 000 pixels along one axis.

The image dataset was divided into training set and test set. The training set consisted of approximately 80 percent of the images, 195 WSIs in total. The test set included the remaining 75 WSIs.

3.3 Implementation

The implemented image analysis system was done using python programming language. Interactive programming environment Spyder (Scientific PYthon Development EnviRonment) was used in the implementation. The implemented system is a modification from method presented by Valkonen et al. [11; 12]. The main additions include the multiscale approach and deep CNN features. Moreover, the set of engineered features is slightly different, since the first version was implemented using Matlab software.

The image analysis workflow has two different training phases. First training phase is performed for the CNN network. After the CNN network is trained, it can be used to extract deep CNN features. The second training phase is performed for the final model that utilises both, deep CNN features and engineered features to derive the image interpretation. As an output, the final model provides confidence map presenting the probability of certain tissue region within each test WSI to contain metastatic tissue. Figure 3.2 presents an overview of the implemented image analysis system. The architecture of the CNN is presented in Figure 3.3.

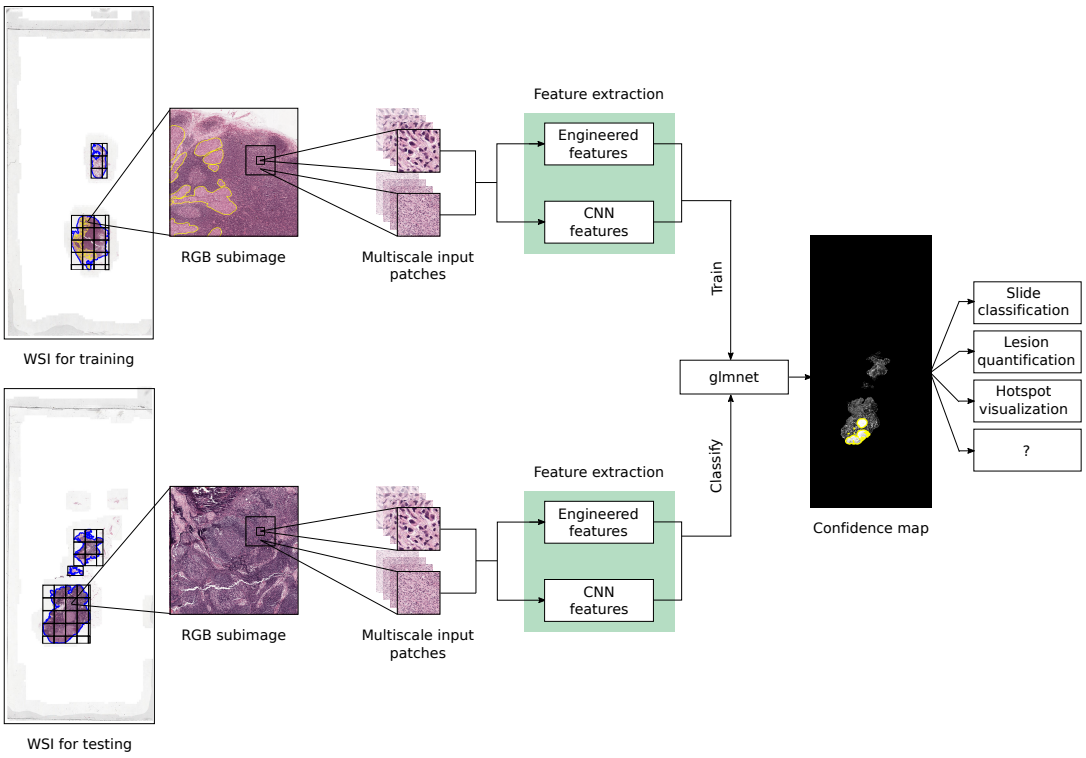

Figure 3.2 *Image analysis system workflow. The upper half presents the training of the model and the lower half presents the steps for classifying a new image with the trained model. The input images include multiscale perspective of the tissue. The feature extraction module consists of two parts, extraction of engineered features and extraction of deep CNN features. As an output, the model provides confidence map presenting the probability of certain tissue region to contain metastatic tissue.*

3.3.1 Preprocessing

The preprocessing of the WSIs was implemented as presented in Valkonen et al. [11]. In order to simplify the classification task and to reduce the amount of data,

first a rough segmentation step was performed for each image to detect the lymph node tissue while excluding the background and most of the adipose tissue.

The visual appearance of the two independent image sets is quite notable due to different scanner and staining protocols. Histogram matching was applied to correct this color variation across the WSIs. Histogram matching was applied separately to each color channel. The training image Tumor 015.tif was selected as the reference based on visual examination.

For convenient handling of the image data during model training and classification, the images were divided into smaller subimages and stored in JPEG2000 format, along with information of their location within the WSI. Each resulting subimage had approximately dimensions of 8192 x 8192. The tissue segmentation and ground truth masks were processed similarly and saved in TIF format.

Additionally, color deconvolution [33] was applied before extracting engineered features. In this study, scikit-image [56] implementation of the color deconvolution algorithm was used. The scikit-image is a collection of algorithms for image processing in Python. The deep CNN features were extracted from RGB images.

3.3.2 Training of the convolutional neural network

Keras module [57] was used to build two parallel sequential models merged into one model. The Figure 3.3 presents the CNN architecture that was implemented for this image analysis system. The networks takes two RGB images as input: patch of 128 x 128 image blocks and patch of 640 x 640 image blocks. Both of the input images were downsampled into dimensions of 32 x 32 x 3. The smaller block covers approximately a $31\mu m$ physical area of the tissue and the wide neighborhood covers a $155\mu m$ area of the tissue. Only samples with 50% coverage of the tissue mask or tumor mask within the smaller scale block were accepted for training.

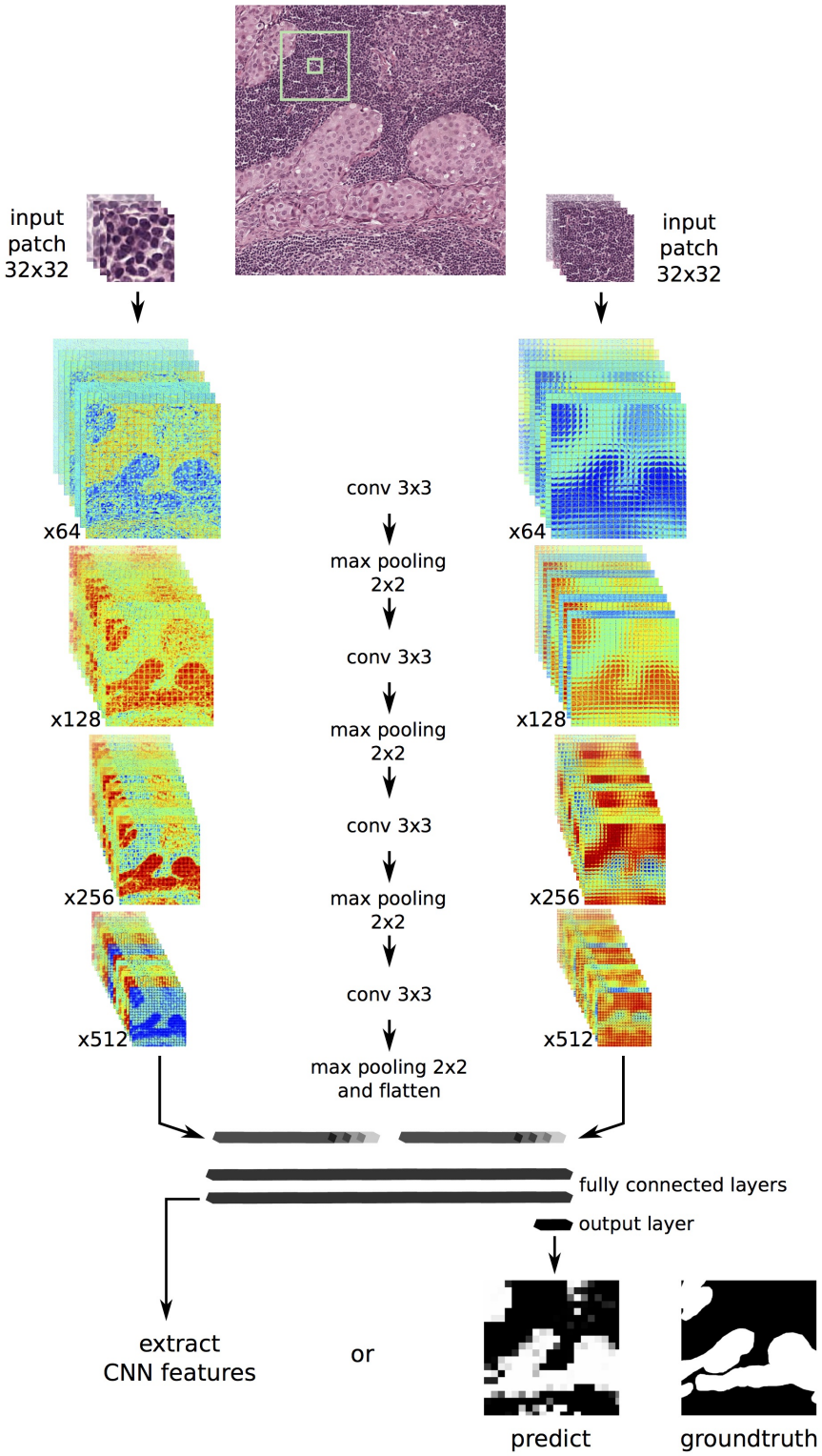

Figure 3.3 *The CNN architecture that was implemented for the image analysis system. The CNN is composed of two identical and parallel convolutional neural networks that are merged into one model and stacked with two fully connected layers.*

Both of the parallel networks include four two-dimensional convolutional layers followed by rectified linear unit (relu) activation and max pooling layer. The kernel size for each convolutional layer was 3 x 3 and 2 x 2 for max pooling. These two parallel networks are merged together with two fully connected layers of length 256 followed by relu activation. The output layer consists of one fully connected output node with sigmoid activation. The total number of trainable parameters in the network is 3758337.

Figure 3.4 presents the two activation functions that were used in the CNN model. Logistic sigmoid function is often used as activation function in neural networks as it maps values from $[-\infty\ \infty]$ to the range [0 1]. This is particularly useful as the activation function in the final output layer, since the function output can be directly used as degree of certainty for each sample to belong to a certain class. The rectifier is the most common activation function used in convolutional neural networks. Rectifier activation is computationally efficient and discovered to accelerate the convergence of stochastic gradient descent in deep models. Additionally, the rectifier is a sparse activation function as it only activates the neurons with non-zero input ($f(x) = max(0, x)$).

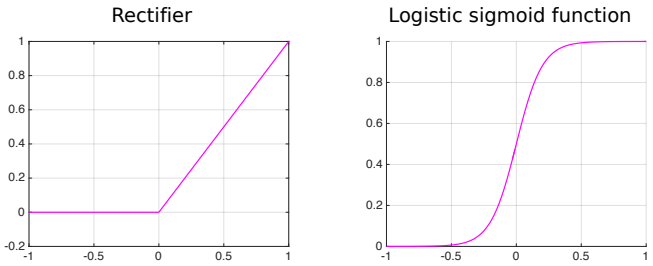

Figure 3.4 *Two common activation functions in convolutional neural networks: rectifier and logistic sigmoid function. Rectifier is a sparse function that activates only non-zero nodes,* $f(x) = max(0, x)$. *Logistic sigmoid function maps values from* $[-\infty\ \infty]$ *to the range* [0 1].

Binary cross-entropy was used as a loss function and the optimization procedure was performed using stochastic optimization algorithm Adam [58]. For Adam algorithm, step size was set to 0.001, exponential decay rates for the moment estimates (β_1 and β_2) were set to 0.9 and 0.999, and the used fuzz factor was $1 * 10^{-8}$. Learning rate decay over each update was set to 0.

The training set was divided into 12 sets to prevent the model from overfitting

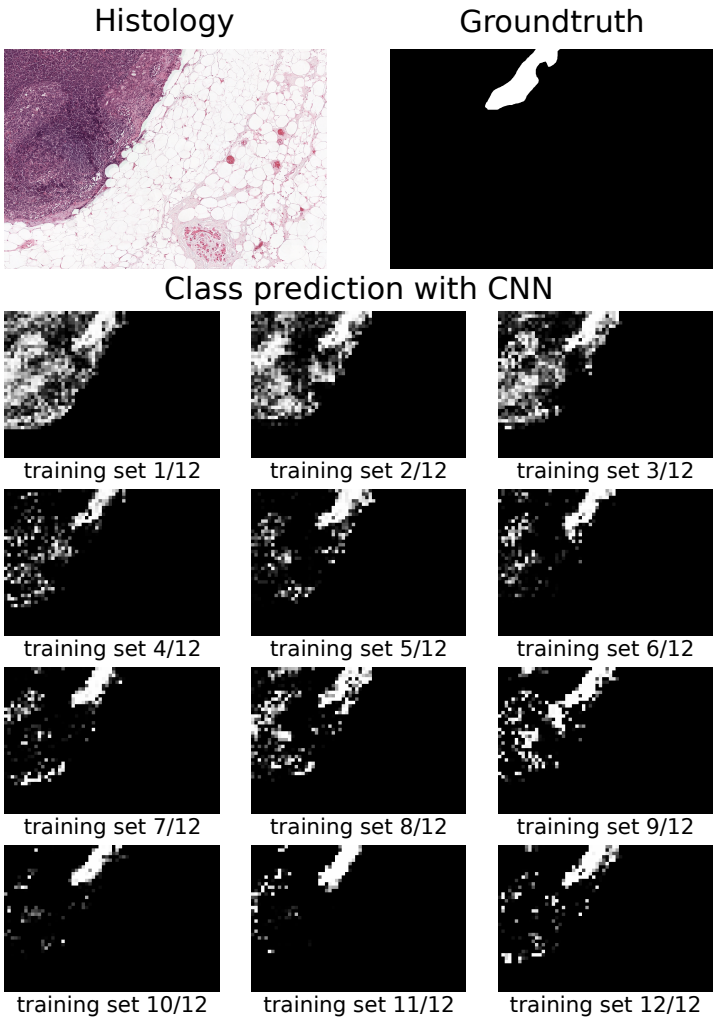

Figure 3.5 *The performance progress of the CNN model after each training set.*

to one training set. The performance progress of the CNN model is presented in Figure 3.5. The figure presents classification confidence of one tissue subimage after training with each of the 12 training sets. Each training set included same proportion of images from both of the independent datasets. The whole dataset included more normal tissue area than metastatic tissue. Therefore, to balance the training data, random sampling of the normal tissue samples was done to get even number of tumor samples and normal tissue samples. Each set of the 12 training sets was used to train the CNN model for 5 epochs with a batch size of 100 samples. The model was trained for 60 epochs, and in total, with approximately 6.4 million samples. The training accuracy and loss are presented in Figure 3.6.

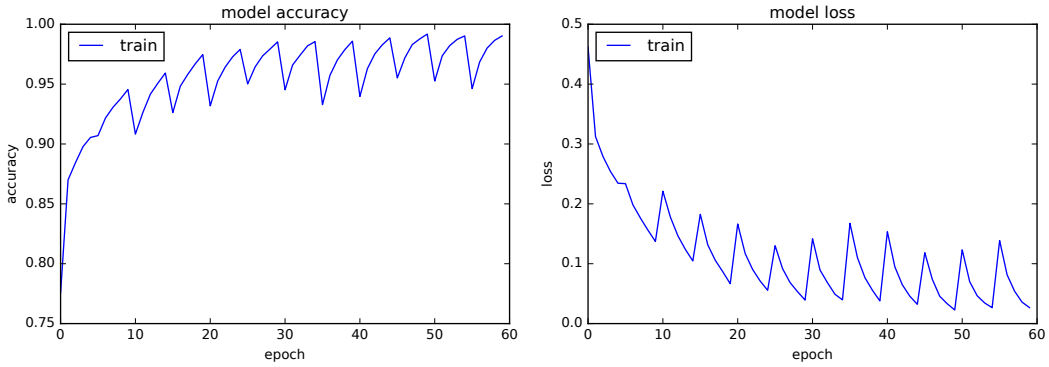

Figure 3.6 *The training accuracy and loss obtained after training the CNN model with 12 training sets, 5 epochs each.*

3.3.3 Feature extraction

The feature extraction step consists of two parts as presented in Figure 3.2: extraction of engineered features and extraction of deep CNN features. Input image is processed blockwise in a way that each 128 x 128 block is presented with vector of 1344 feature values. These features include 1088 engineered features and 256 deep CNN features extracted from tissue image with a multiscale perspective. Thus, the quantitative representation of a 128 x 128 pixel block includes also the information about its neighboring area.

Extraction of engineered features

The feature extraction module for engineered features includes 272 different features in total. The list of engineered features is presented in Table 3.1. This same feature extraction module was applied four times, hematoxylin and eosin channel of two tissue blocks with different scale. In total, one 128x128 pixel block can be presented with 1088 engineered features. Scikit-image implementations were used to extract these engineered features. Similar set of features with minor variations has been successfully used in different image analysis applications, such as, detecting metastasis from whole slide images [11], and detecting early neoplastic changes [12; 59].

The extracted features included first- and second-order statistical texture features

derived from image histogram and GLCM, such as, mean intensity, variance, skewness, and angular second moment. These co-occurring gray level value properties were calculated at offset distance of one pixel and with respect to four angles: $[0°, 90°, 180°, 270°]$.

Additionally, features were extracted using local binary patterns (LBP). The circularly symmetric neighborhood is determined by assigning parameters that control the quantization of the angular space and radius of the neighborhood. For the implemented method, the LBP responses were calculated for two different circularly symmetric neighborhood: radius of 3 pixels and angular space of 24 points, and radius of 5 pixels and angular space of 40 points. The response image to these both LBP operators over a sample block present the uniform patterns in the sample block. Properties of discrete occurrence histogram of these response images were used as features. In addition, HOG features were extracted. First, gradient image is computed and then histograms of gradient orientation angles are computed within smaller cells of the image. Finally, the orientation histograms of all cells are flattened into a feature vector. Also daisy features were extracted with a number of circles set to 2. From each circle 6 histograms were calculated with 8 orientation bins.

Extraction of deep CNN features

The final dense layer before output layer in the merged CNN architecture was used as deep CNN feature vector. This is visualised in 3.3. In total, one 128 x 128 pixel block was represented with 256 deep CNN features. This feature vector includes the multiscale perspective.

3.3.4 Classification

Classification was performed for the test set of 75 WSIs. Four different models were trained to obtain the contribution of different type of features. First, the trained CNN model was used for classification instead of extraction of features. The other three models were build on glmnet model, one trained with only engineered features, one with only deep CNN features, and one with engineered features and deep CNN features. For all these four models, the probability to belong to the group of metastatic tissue was predicted to get a confidence heatmap of each test tissue image. In addition, a class was predicted for each block within the test subimages to

Table 3.1 *Amount of different type of features. The engineered features include texture features and LBP, HOG, and Daisy descriptors. Since each engineered feature was extracted from hematoxylin and eosin channel with multiscale approach, the total amount of features is four times the number of different features. Deep CNN features were extracted from RGB images and this feature vector already includes the multiscale perspective. In total, 1344 features were extracted.*

Feature type	number of features	color channel	total amount of features
Texture feature	40	H&E	160
LBP feature	32	H&E	128
HOG feature	96	H&E	384
Daisy feature	104	H&E	416
Deep CNN feature	256	RGB	256
			1344

get a class mask. After the confidence map and class prediction for each subimage is obtained with each model, the confidence subimages and class masks are stitched back into WSI images.

The CNN model predicts only the probability of each class. To use the CNN model for classification, a threshold was applied to the output confidence image. The threshold value was set to 0.5, thus, with confidence < 0.5 the predicted class is 0 (normal tissue) and with with confidence $>= 0.5$ the predicted class is 1 (metastatic tissue).

Python package for fitting generalised linear model (glmnet [52]) was used in this study. Three different logistic regression models with different set of features were trained. Foe each model, the α was set to 0.5, therefore, the elastic net regularization uses both lasso and ridge. The glmnet algorithm fits series of models to determine the λ value that controls the overall strength of the regularisation. In total, 100 values for λ were computed. After computing the path of λ values, the performance of the model is analysed using 3-fold cross validation. The value of λ which achieved the best performance in cross validation was used in prediction. For model with engineered features, the best value for λ was 0.0030. For the model with deep CNN features, the best value for λ was 0.1903. And for the model with both deep CNN features and engineered features, the best value for λ was 0.0002.

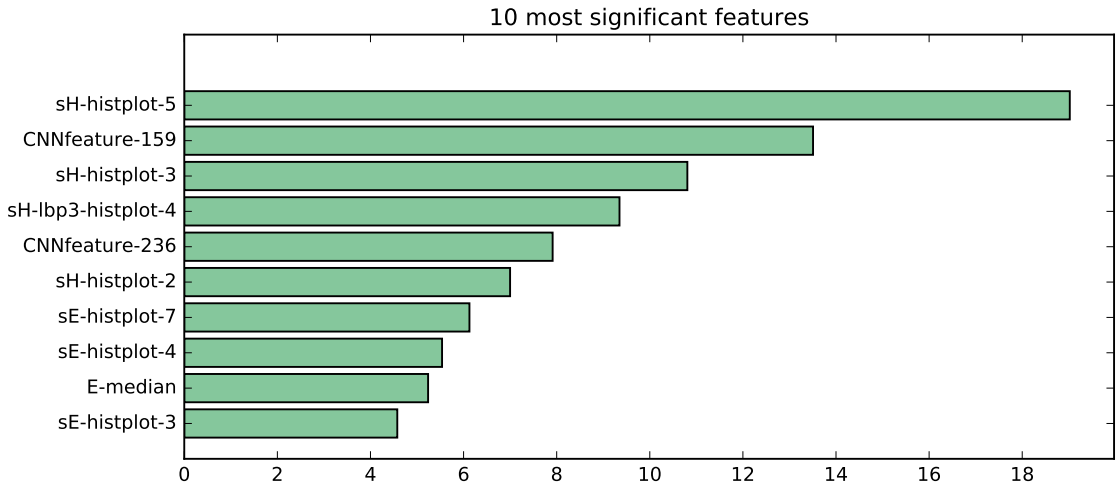

Figure 3.7 *The coefficients of 10 most significant features related to the classification process of glmnet model trained with deep CNN features and engineered features.*

The feature data was normalised to the range [0 1]. The number of training samples for each model was approximately 100 000 positive samples and 100 000 negative samples. The samples for training the final glmnet model were randomly sampled from all the available data. However, this was done in a way that the train set included samples from all WSIs within the training set. In total, the train set included 773 subimages containing metastatic tissue and 10 404 subimages of normal tissue. Consequently, 130 sample blocks were selected from each tumor subimage and 10 sample blocks from each normal subimage.

The glmnet model also provides coefficients related to the significance of each feature with respect to the classification. The 10 most significant features for glmnet model trained with both deep CNN features and engineered features is presented in Figure 3.7.

4. RESULTS

The implemented image analysis system was evaluated with test set of 75 WSIs of lymph node sections. Each test WSI was scored with confidence levels using four different model, each trained with 195 WSIs.

Examples of class predictions performed by each model are presented in Figure 4.1. First model was the trained CNN model that was used for classification. The other three models were glmnet models trained with different set of features: one with only engineered features, one with only deep CNN features and one with both engineered features and deep CNN features. The example images include five 8192 x 8192 subimages of the tissue, the corresponding tumor annotations, and confidence maps generated by each model. Blockwise ROC curve was calculated each example image and the corresponding AUC measures are also presented in the Figure 4.1.

Table 4.1 *Classification performances of each model measured with percentage of correctly classified samples, sensitivity, specificity, and f-score. Glmnet+ef stands for the glmnet model trained with engineered features. Glmnet+CNNf stands for the glmnet model trained with deep CNN features. CNN prediction row presents the results for using the CNN as a classifier. And finally, glmnet+CNNf+ef stands for the glmnet model that was trained with both engineered features and deep CNN features.*

Model	Correctly classified samples (%)	Sensitivity (%)	Specificity (%)	F-score
Glmnet+ef	87.9	87.6	93.4	0.93
Glmnet+CNNf	87.7	87.6	87.7	0.93
CNN prediction	92.8	94.0	70.9	0.96
Glmnet+CNNf+ef	91.3	91.2	91.7	0.95

In order to evaluate the performance of each model also numerically, all of the confidence values were collected and blockwise ROC curves were calculated for each

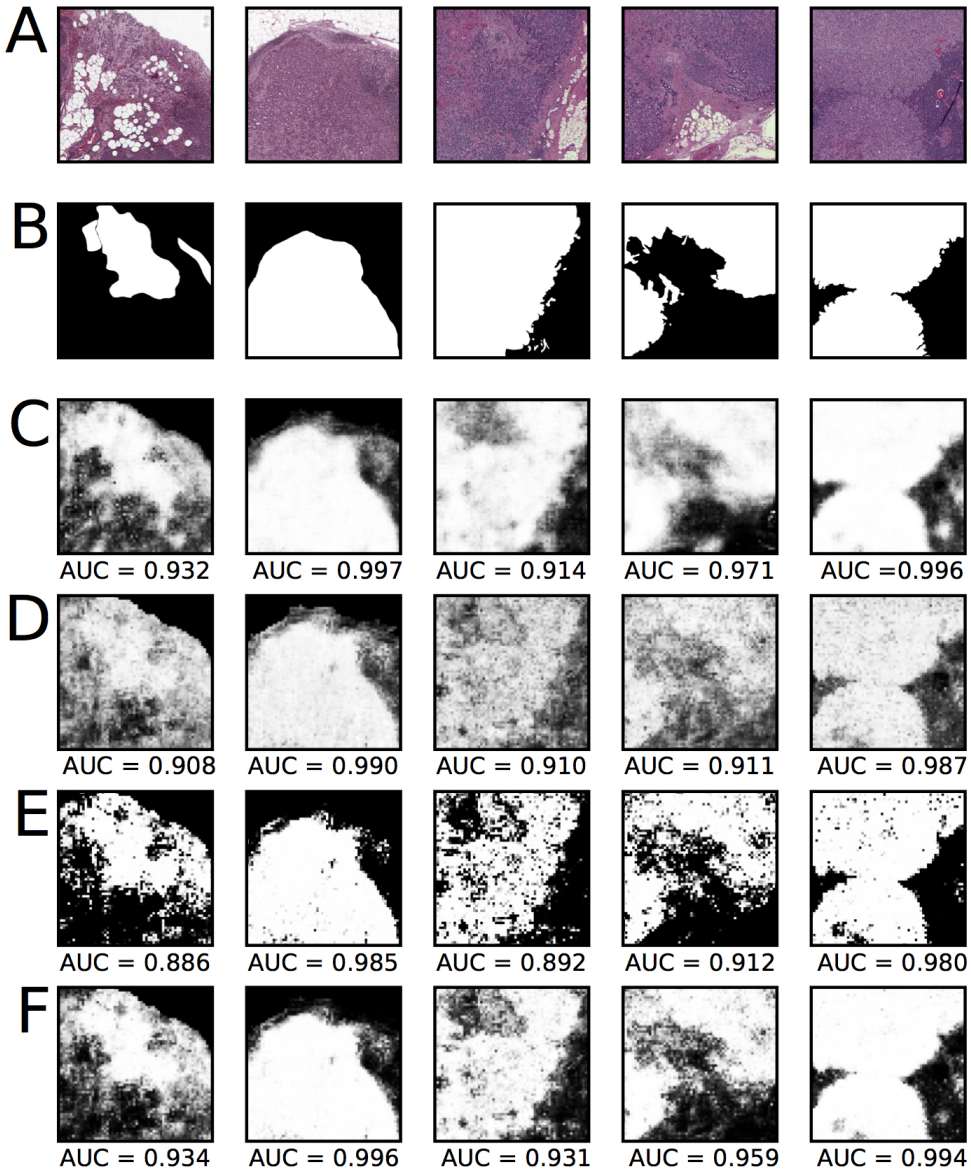

Figure 4.1 *Confidence maps predicted for five different tissue subimage. The tissue histology for each color corrected example is presented in the row* ***A****. The row* ***B*** *presents the groundtruth annotation, where mask area stands for metastatic tissue. The class predictions are generated using four different classification model: glmnet model using engineered features (****C****), glmnet model using deep CNN features (****D****), the CNN classifier (****E****), and glmnet model using deep CNN features and engineered features (****F****). The confidence denote the probability of a certain tissue block to belong to the group of cancerous tissue. Also ROC curve was calculated for each confidence map and the corresponding AUC value is presented below each confidence map.*

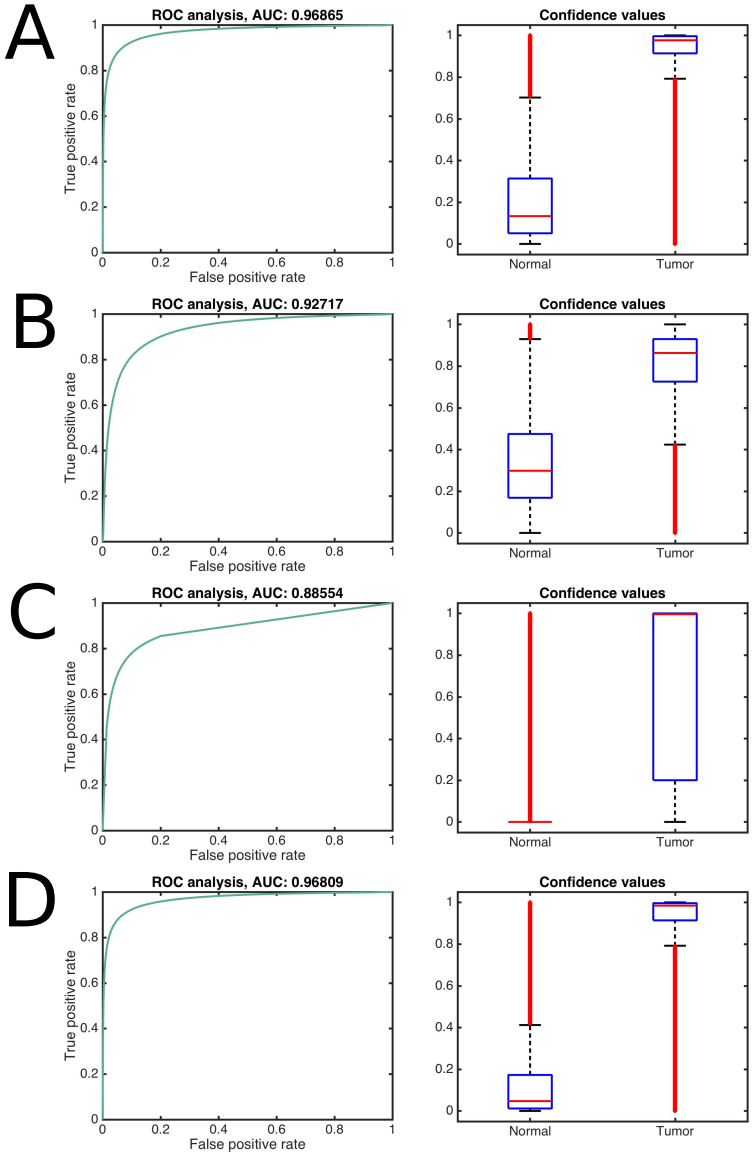

Figure 4.2 *All of the confidence values from the test set predicted with each model were collected and blockwise ROC curve and AUC measure were calculated. Further, the variation of confidence values gathered from metastatic tissue blocks and normal tissue blocks are visualised in the boxplots. On each box, the central mark presents the median value, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The outliers are plotted individually using the symbol. Row* ***A*** *presents the results for glmnet model using engineered features,* ***B*** *for glmnet model using deep CNN features,* ***C*** *for the CNN classifier, and* ***D*** *for glmnet model using deep CNN features and engineered features.*

model. These ROC curves are presented in Figure 4.2. Only tissue blocks within the tissue mask area were included in the ROC analysis. The obtained AUC measure for CNN classifier was 0.894, 0.934 for the glmnet model with deep CNN features, 0.975 for the glmnet model with engineered features, and 0.974 for the glmnet model with deep CNN features and engineered features. Further, the variation of confidence values of metastatic tissue blocks and normal tissue blocks were visualised with boxplots. These boxplots are also presented in Figure 4.2. On each box, the central mark presents the median value, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The dashed line extend to the last data points that are not considered as outliers. The outliers are plotted individually using the symbol.

In total, the test set included approximately 420 000 positive sample blocks from tumor area and 7 900 000 sample blocks from normal tissue area. In total, the test set included approximately 8.3 million sample blocks. Percentage of correctly classified samples, classification sensitivity and specificity, and f-score were calculated for each different model. These results are presented in Table 4.1. The percentage of correctly classified samples was $92,9\%$ for CNN classifier, $87,9\%$ for glmnet classifier trained with engineered features, $87,7\%$ for glmnet classifier trained with deep CNN features, and $91,3\%$ for glmnet classifier trained with deep CNN features and engineered features.

5. DISCUSSION

The results of the study show that the model provides an accurate detection of hot-spot regions in WSIs. This is visualised in Figure 5.1, that shows an example of a classification result for a WSI from both of the datasets. In the figure, an original WSI with tumor annotations and the corresponding confidence values given by the implemented model are shown. The confidence maps are predicted with the glmnet model that was trained using both engineered features and deep CNN features. The higher confidence values are concentrated in areas marked as tumor by the pathologist, while confidences in normal tissue area are mainly lower.

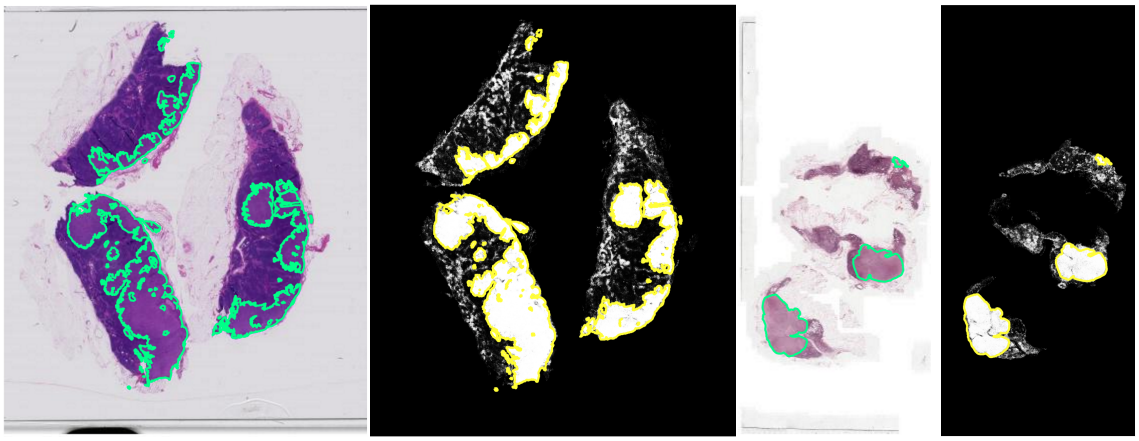

Figure 5.1 *The confidence maps for two example WSIs predicted using the glmnet model that was trained using both engineered features and deep CNN features.*

The numerical evaluation of the implemented method is presented in Table 4.1. The table lists the classification performance using four different classifiers: the glmnet classifier trained with three different set of features, and the CNN model. The percentage of correctly classified samples is quite high for each case. When considering only the percentage of correctly classified samples, the CNN classifier reaches the best accuracy, as was expected based on the literature. The second best classification performance is obtained with the glmnet model that uses both deep CNN features and engineered features. However, when sensitivity and specificity

measures and f-score are also taken into account, the best model is the glmnet model that uses both deep CNN features and engineered features.

Notable observation from these numerical results is the performance of the CNN model. The deep CNN features combined with glmnet model has the lowest percentage of correctly classified samples. Yet, the CNN classifier reaches the highest percentage of correctly classified samples. This seems to indicate that the glmnet is most likely not trained to its full potential. Only few sets of parameters were tested for the glmnet model, therefore, it is most likely that with exhaustive parameter optimization, the performance of the glmnet model could be better. In addition, the low amount of deep CNN features might also have an effect on the model performance.

The numerical evaluation of the implemented method included also blockwise ROC analysis. Confidence values predicted with each model were collected from the whole test set and analysed. The results of ROC analysis and corresponding AUC values are presented in Figure 4.2 for each model. Variation of these confidence values between normal tissue and metastatic tissue are visualised with boxplots in Figure 4.2. It can be observed, that the best results are achieved again with the glmnet model with both engineered features and deep CNN features. This model reaches the best AUC value and also the confidence values in the boxplot are most distinct between the two classes. However, the AUC value of glmnet with engineered features is just slightly better, yet, the confidence values are not as distinct in the boxplot.

The worst AUC measure is obtained using the CNN classifier. This is understandable since the CNN classifier predicts mainly really high confidence values or values close to zero. This of course, can be directly seen in the ROC analysis that is based on changing the discrimination threshold of the classifier. This is also distinct in the boxplot (Figure 4.2C) where the central mark presents the median value of the class confidence values, for tumor class the mark is at one and for normal class the mark is at zero. The reason for this low AUC measure is most likely the amount of false positives. Only samples with 50% coverage of the tissue mask or tumor mask within the smaller scale block were accepted for training. This has probably caused misclassifications in the border of healthy tissue and cancerous tissue. This could have been improved for example by using positive training samples with 100% coverage of the tumor mask and negative training samples with 50% coverage of the tissue mask.

Considering all of the performance measures, the results seem to indicate that the CNN model is overfitting. This effect can be seen from the training progress of the network in Figures 3.5 and 3.6. After the network has seen three training sets of the total of 12, no significant progress can be seen. The network overfits to one set and after new set is presented, the training accuracy drops. Nevertheless, since the CNN does not include any regularization layers, such as, dropout, these results are as expected. Therefore, it is not unexpected that deep CNN features combined with glmnet with regularization reaches better AUC value than CNN classifier alone.

In addition to hot-spot detection provided by the confidence maps, the method offers tools for objective and quantitative characterisation of tissue histology. The implemented feature extraction library includes 1344 features in total, and combined with machine learning model, an application specific set of significant features can be obtained. The 10 most significant features related to metastasis detection from lymph node sections is presented in Figure 3.7. This list of features is obtained from the glmnet model that was trained with engineered features and deep CNN features. These top ten most important features contributing to the classifier model are in practice the descriptors which behave differently in normal and metastatic tumor tissue areas. The list consists of two deep CNN features and eight engineered features. Letter H and E in the name of engineered feature refers to the image channel that the feature has been extracted from: hematoxylin channel or eosin channel. The letter s refers to the scale of the tissue block from where the feature is extracted from. If there is the letter s the feature is extracted from the wider neighborhood, otherwise the feature is extracted from the smaller neighborhood. Most of the features are gray level histogram properties, however, one of the selected features is a property of uniform pattern occurrence histogram obtained from LBP operation response.

While some of the significant features are not straightforwardly interpretable, most of the features support existing prior knowledge about the difference of tissue histology in normal and metastatic tissue and are easy to interpret. For example, the most important feature that was selected for this model is a histogram property of hematoxylin channel. This supports the knowledge that normal tissue contains more hematoxylin than metastatic tissue, this is visualised in Figure 5.2. There is distinctly more blue hematoxylin in normal sample than in the metastatic tissue sample. This, of course would be clearly visible in the histogram of the image sample.

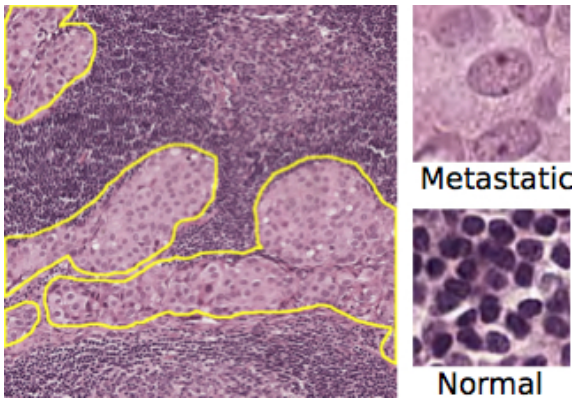

Figure 5.2 *Tissue histology examples from normal tissue area and from metastatic tissue. The yellow annotation surrounds the tumor area.*

The evaluation of the results is not an easy and straightforward task. From classification point of view, certain measures can be calculated to evaluate the performance of the implemented method. However, from the pathologist's viewpoint, these measures don't necessarily give real accuracy of the method. This type of blockwise evaluation gives more weight on larger tumor regions in the final scoring, since they consist of a larger number of pixels than smaller lesions. For the pathologist, this is a problem, since large macrometastases can often be spotted more easily than the small ones.

If a realistic usage of the implemented method is considered, it would be used as a decision support system, not as a tool for diagnosis. In this case, the high sensitivity of the method and the confidence map output show a potential that the method could be used in the future as a software for computer aided diagnosis. High sensitivity is useful for ruling out tumorous tissue, since 100 % sensitivity would mean that the method recognises all of the areas with metastatic tissue. The workload of a pathologist could be reduced if an automated image analysis system with close to 100 % sensitivity could be used to pre-screen histological slides and to exclude even part of the slides.

Nevertheless, the method presented here still does not reach the point of 100 % sensitivity. Even though, the ROC curve seems to reach a point of 100 % sensitivity with a certain threshold, the boxplot reveals that some tumor samples still score low confidence values and therefore are left undetected. However, progress has been made when comparing to the results presented by Valkonen et al. [11]. The combination of engineered features and deep CNN features reached the AUC value

of 0.97 while similar method presented by Valkonen et al. [11] reached AUC value of 0.90. However, the results are not directly comparable since the other result includes confidence score for the whole dataset of 270 WSIs, while this study used only test set of 75 WSIs.

All things considered, the combination of deep CNN features and engineered features yields to a more accurate model that either one alone. Yet, the full potential of deep learning and deep CNN features was not achieved in this study. Therefore, there is still need for further research related to deep learning and convolutional neural networks in bioimage analysis. In addition, the optimization of glmnet parameters was very minimal in this study, therefore, the potential of the implemented method is prominent and this motivates and encourages for further research. The aim of this study was to develop methods for quantitative characterisation of tissue histology, which was achieved here.

6. CONCLUSIONS

The aim of this thesis study was to implement an image analysis pipeline for analysing histological images and quantitatively characterising tissue histology. To achieve this, a feature extraction library was implemented that utilises feature engineering and convolutional neural network to automatically extract quantitative characteristics from histological images. Machine learning methods with supervised learning were then applied to train multiple discriminative models based on these quantitative histological characteristics. The performance of the implemented method was evaluated in detection of breast cancer metastases in lymph node sections from HE stained whole slide images.

The results of the study, show that the implemented image analysis system was able to accurately separate metastatic tissue from normal tissue (AUC = 0.89–0.97). Therefore, the implemented image analysis system provides an automated detection of hot-spot regions in WSIs. In addition, a set of significant features related to the classification task can be obtained from the presented method. This provides insight to the quantitative characteristics of tissue histology that separate metastatic tissue from normal tissue. In all, the implemented method is modular and therefore, it can be easily extended with new features. Since the feature extraction library is generic, it makes the method generalizable to a variety of bioimage analysis problems.

This research was performed at BioMediTech, University of Tampere, and it is part of three-dimensional (3D) histogenomic modeling of whole prostate project funded by TEKES. The research related to quantitative characterization tissue histology will continue and the developed feature extraction library will be further extended to three dimensions and integrated into 3D histogenomic model of prostate that is being currently developed. Further, implementations for general image analysis tools for bioimage analysis, such as, histogram matching and color deconvolution were also developed. All of these implemented image analysis tools can be directly integrated into software that can reduce the workload of pathologists by guiding the

pathologist to concentrate on regions of interest, and providing tools to quantify tissue histology. These advancements will help to move towards the transition to digital pathology in clinical practice.

REFERENCES

[1] H. Peng. Bioimage informatics: A new area of engineering biology. Bioinformatics, Vol. 24, No. 17, 2008. pp. 1827–1836.

[2] G. Danuser. Computer vision in cell biology. Cell, Vol. 147, No. 5, 2011. pp. 973–978.

[3] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri. NIH Image to ImageJ: 25 years of image analysis. Nature Methods, Vol. 9, No. 7, 2012. pp. 671.

[4] R. D. da Silva, R. Minetto, W. R. Schwartz, and H. Pedrini. Satellite image segmentation using wavelet transforms based on color and texture features. Proceedings of International Symposium on Visual Computing, 2008, pp. 113–122.

[5] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, 2005, pp. 568–573.

[6] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat. A survey on industrial vision systems, applications and tools. Image and Vision Computing, Vol. 21, No. 2, 2003. pp. 171–188.

[7] F. Ghaznavi, A. Evans, A. Madabhushi, and M. Feldman. Digital imaging in pathology: Whole-slide imaging and beyond. Annual Review of Pathology: Mechanisms of Disease, Vol. 8, 2013. pp. 331–359.

[8] M. Veta, J. P. Pluim, P. J. Van Diest, and M. A. Viergever. Breast cancer histopathology image analysis: A review. IEEE Transactions on Biomedical Engineering, Vol. 61, No. 5, 2014. pp. 1400–1411.

[9] S. Al-Janabi, A. Huisman, and P. J. Van Diest. Digital pathology: Current status and future perspectives. Histopathology, Vol. 61, No. 1, 2012. pp. 1–9.

[10] A. Madabhushi and G. Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities. 2016.

[11] M. Valkonen, K. Kartasalo, K. Liimatainen, M. Nykter, L. Latonen, and P. Ruusuvuori. Metastasis detection from whole slide images using local features and random forests. Cytometry Part A, 2017.

[12] M. Valkonen, P. Ruusuvuori, K. Kartasalo, M. Nykter, T. Visakorpi, and L. Latonen. Analysis of spatial heterogeneity in normal epithelium and preneoplastic alterations in mouse prostate tumor models. Scientific Reports, Vol. 7, 2017.

[13] A. H. Beck, A. R. Sangoi, S. Leung, R. J. Marinelli, T. O. Nielsen, M. J. van de Vijver, R. B. West, M. van de Rijn, and D. Koller. Systematic analysis of breast cancer morphology uncovers stromal features associated with survival. Science translational medicine, Vol. 3, No. 108, 2011. pp. 108ra113–108ra113.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, Vol. 521, No. 7553, 2015. pp. 436–444.

[16] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Transactions on Medical Imaging, Vol. 35, No. 5, 2016. pp. 1196–1206.

[17] M. Sonka, V. Hlavac, and R. Boyle. Image processing, analysis and machine vision. Springer, 1993.

[18] W. K. Pratt. Digital image processing: PIKS scientific inside. Wiley-Interscience, USA, 2007.

[19] F. S. Abas, H. N. Gokozan, B. Goksel, J. J. Otero, and M. N. Gurcan. Intraoperative neuropathology of glioma recurrence: Cell detection and classification. Proceedings of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2016,

[20] S. Doyle, M. Hwang, K. Shah, A. Madabhushi, M. Feldman, and J. Tomaszeweski. Automated grading of prostate cancer using architectural and textural image features. Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007, pp. 1284–1287.

[21] R. Turkki, N. Linder, P. E. Kovanen, T. Pellinen, and J. Lundin. Identification of immune cell infiltration in hematoxylin-eosin stained breast cancer samples: Texture-based classification of tissue morphologies. Proceedings of SPIE Medical Imaging, 2016, pp. 979110–979110.

[22] V. J. Tuominen, S. Ruotoistenmäki, A. Viitanen, M. Jumppanen, and J. Isola. ImmunoRatio: A publicly available web application for quantitative image analysis of estrogen receptor (ER), progesterone receptor (PR), and Ki-67. Breast Cancer Research, Vol. 12, No. 4, 2010. pp. R56.

[23] J. Diamond, N. H. Anderson, P. H. Bartels, R. Montironi, and P. W. Hamilton. The use of morphological characteristics and texture analysis in the identification of tissue composition in prostatic neoplasia. Human Pathology, Vol. 35, No. 9, 2004. pp. 1121–1131.

[24] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Adaptive Computation, Machine Learning. Cambridge, Massachusetts: The MIT Press, 2016.

[25] K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, et al.. Gland segmentation in colon histology images: The glas challenge contest. Medical Image Analysis, Vol. 35, 2017. pp. 489–502.

[26] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck. Deep learning for identifying metastatic breast cancer. arXiv preprint arXiv: 1606.05718, 2016.

[27] R. Chen, Y. Jing, and H. Jackson. Identifying metastases in sentinel lymph nodes with deep convolutional neural networks. arXiv preprint arXiv: 1608.01658, 2016.

[28] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. Nature, Vol. 542, No. 7639, 2017. pp. 115–118.

[29] H. Wang, A. Cruz-Roa, A. Basavanhally, H. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. Gonzalez, and A. Madabhushi. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. Journal of Medical Imaging, Vol. 1, No. 3, 2014. pp. 034003–034003.

[30] G. Li and Y. Yu. Visual saliency based on multiscale deep features. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5455–5463.

[31] A. A. A. Setio, A. Traverso, T. de Bel, M. S. Berens, C. v. d. Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, et al.. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. arXiv preprint arXiv: 1612.08012, 2016.

[32] R. C. Gonzalez and R. E. Woods. Digital image processing. Prentice hall Upper Saddle River, NJ, USA, 2008.

[33] A. C. Ruifrok, D. A. Johnston, et al.. Quantification of histochemical staining by color deconvolution. Analytical and Quantitative Cytology and Histology, Vol. 23, No. 4, 2001. pp. 291–299.

[34] D. Swinehart. The beer-lambert law. J. Chem. Educ, Vol. 39, No. 7, 1962. pp. 333.

[35] J. van der Laak, M. M. Pahlplatz, A. G. Hanselaar, and P. de Wilde. Hue-saturation-density (HSD) model for stain recognition in digital images from transmitted light microscopy. Cytometry, Vol. 39, No. 4, 2000. pp. 275–284.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. John Wiley & Sons, 2012.

[37] A. Materka, M. Strzelecki, et al.. Texture analysis methods–a review. Technical university of lodz, Institute of Electronics, COST B11 report, Brussels, 1998. pp. 9–11.

[38] M. Sikiö. Textural features in medical magnetic resonance image analysis of the brain and thigh muscles. Tampere University of Technology, PhD Thesis, 2016.

[39] B. Julesz, E. Gilbert, L. Shepp, and H. Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics—revisited. Perception, Vol. 2, No. 4, 1973. pp. 391–405.

[40] R. M. Haralick, K. Shanmugam, et al.. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, No. 6, 1973. pp. 610–621.

[41] T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. Proceedings of European Conference on Computer Vision, 2000, pp. 404–420.

[42] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, 2002. pp. 971–987.

[43] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[44] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 5, 2010. pp. 815–830.

[45] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, Vol. 60, No. 2, 2004. pp. 91–110.

[46] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, 2005. pp. 1615–1630.

[47] S. S. Haykin. Neural networks: A comprehensive foundation. Tsinghua University Press, 2001.

[48] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology, Vol. 160, No. 1, 1962. pp. 106–154.

[49] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. 1982. pp. 267–285.

[50] T. M. Mitchell. Machine learning. McGraw-Hill Boston, MA, 1997.

[51] C. M. Bishop. Pattern recognition and machine learning. Springer, 2006.

[52] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, Vol. 33, No. 1, 2010. pp. 1.

[53] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 67, No. 2, 2005. pp. 301–320.

[54] T. Fawcett. An introduction to ROC analysis. Pattern Recognition Letters, Vol. 27, No. 8, 2006. pp. 861–874.

[55] B. Ehteshami Bejnordi and J. van der Laak. CAMELYON16: Grand challenge on cancer metastasis detection in lymph nodes. Accessed: 2017-04-05. Available at: https://camelyon16.grand-challenge.org.

[56] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. Scikit-image: Image processing in Python. PeerJ, Vol. 2, 2014. pp. e453.

[57] F. Chollet. Keras. Accessed: 2016-09-01. Available at: https://github.com/fchollet/keras.

[58] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412.6980, 2014.

[59] P. Ruusuvuori, M. Valkonen, M. Nykter, T. Visakorpi, and L. Latonen. Feature-based analysis of mouse prostatic intraepithelial neoplasia in histological tissue sections. Journal of Pathology Informatics, Vol. 7, 2016.